FILEID**FORWRITSO

```
FFFFFFFFFF    000000    RRRRRRRR    WW      WW    RRRRRRRR    IIIIII    TTTTTTTTTT    SSSSSSSS    000000
FFFFFFFFFF    000000    RRRRRRRR    WW      WW    RRRRRRRR    IIIIII    TTTTTTTTTT    SSSSSSSS    000000
FF          00    00    RR    RR    WW      WW    RR    RR      II          TT        SS        00    00
FF          00    00    RR    RR    WW      WW    RR    RR      II          TT        SS        00    00
FF          00    00    RR    RR    WW      WW    RR    RR      II          TT        SS        00    00
FFFFFFFF    00    00    RRRRRRRR    WW      WW    RRRRRRRR      II          TT          SSSSSS  00    00
FFFFFFFF    00    00    RRRRRRRR    WW      WW    RRRRRRRR      II          TT          SSSSSS  00    00
FF          00    00    RR  RR      WW  WW  WW    RR  RR        II          TT              SS  00    00
FF          00    00    RR   RR     WW  WW  WW    RR   RR       II          TT              SS  00    00
FF          00    00    RR    RR    WWW    WWW    RR    RR      II          TT              SS  00    00
FF            000000    RR    RR    WW      WW    RR    RR    IIIIII        TT        SSSSSSSS    000000
FF            000000    RR    RR    WW      WW    RR    RR    IIIIII        TT        SSSSSSSS    000000

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II            SS
LL                II            SS
LL                II            SS
LL                II          SSSSSS
LL                II          SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS
```

N 10

FOR$WRITE_SO   - entry point for FORTRAN WRITE SEQUENTI 16-SEP-1984 00:07:06   VAX/VMS Macro V04-00   Page   0
Table of Contents

```
0000      1              .TITLE  FOR$WRITE_SO - entry point for FORTRAN WRITE SEQUENTIAL OBJECT-FORMA
0000      2              .IDENT  /1-011/          File: FORWRITSO.MAR  Edit: JAW1011
0000      3
0000      4      ;*******************************************************************************
0000      5      ;*                                                                             *
0000      6      ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                    *
0000      7      ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                     *
0000      8      ;*  ALL RIGHTS RESERVED.                                                       *
0000      9      ;*                                                                             *
0000     10      ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED      *
0000     11      ;*  ONLY IN  ACCORDANCE WITH  THE   TERMS   OF   SUCH  LICENSE  AND WITH THE   *
0000     12      ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY   OTHER     *
0000     13      ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY      *
0000     14      ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE   SOFTWARE IS  HEREBY     *
0000     15      ;*  TRANSFERRED.                                                               *
0000     16      ;*                                                                             *
0000     17      ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE      *
0000     18      ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT      *
0000     19      ;*  CORPORATION.                                                               *
0000     20      ;*                                                                             *
0000     21      ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS      *
0000     22      ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                    *
0000     23      ;*                                                                             *
0000     24      ;*                                                                             *
0000     25      ;*******************************************************************************
0000     26
0000     27
0000     28      ;++
0000     29      ; FACILITY: FORTRAN Support Library - user callable
0000     30
0000     31      ; ABSTRACT:
0000     32
0000     33      ;        This module contains the entry point for the FORTRAN
0000     34      ;        WRITE SEQUENTIAL OBJECT-FORMATTED I/O statement.  It is simply
0000     35      ;        a call to FOR$$IO_BEG with bits in R0 which describe the
0000     36      ;        parameter list.  FOR$$IO_BEG interprets the parameters.
0000     37
0000     38      ; MAINTENANCE NOTE:
0000     39      ;        The transfer vector (RTLVECTOR+ALLGBL) must have the following:
0000     40
0000     41      ;        .TRANSFER       FOR$WRITE_SO
0000     42      ;        .MASK           FOR$$IO_BEG
0000     43      ;        BRW             FOR$WRITE_SO+2
0000     44
0000     45      ;        This puts the correct mask in entry vector, that is FOR$$IO_BEG entry mask.
0000     46      ;        Furthermore this module must only use R0 and R1
0000     47      ;        since any other register might not be in the entry mask for FOR$$IO_BEG.
0000     48
0000     49      ; ENVIRONMENT: User access mode; mixture of AST level or not
0000     50
0000     51      ; AUTHOR:        Richard B. Grove, CREATION DATE: 28-May-78
0000     52
0000     53      ; MODIFIED BY:
0000     54      ;        T. Hastings, 29-July-78
```

FORSWRITE_SO   C 11
1-011   – entry point for FORTRAN WRITE SEQUENTI 16-SEP-1984 00:07:06   VAX/VMS Macro V04-00   Page   2
     HISTORY ; Detailed Current Edit History   6-SEP-1984 11:02:14   [FORRTL.SRC]FORWRITSO.MAR;1   (2)

```
0000    56              .SBTTL  HISTORY         ; Detailed Current Edit History
0000    57
0000    58
0000    59    ; Edit History for Version 1
0000    60    ;
0000    61    ; 0-10  - Add comment about vectors.  TNH 23-June-78
0000    62    ; 0-12  - Pass arg in R0, not ROR, add comments.  TNH 29-July-78
0000    63    ; 1-001 - Update version number and copyright notice.  JBS 16-NOV-78
0000    64    ; 1-002 - Change statement type symbols to be LUB$K...  JBS 07-DEC-78
0000    65    ; 1-003 - Change statement type symbols to be ISB$K...  JBS 11-DEC-78
0000    66    ; 1-004 - Add "A" to the PSECT directive.  JBS 22-DEC-78
0000    67    ; 1-005 - Add FOR$READ_KF, FOR$READ_KO, FOR$REWRITE_SF, FOR$REWRITE_SO,
0000    68    ;          FOR$READ_IF, FOR$READ_IO, FOR$WRITE_IF, FOR$WRITE_IO,
0000    69    ;          FOR$READ_KU, FOR$REWRITE_SU,
0000    70    ;          SBL 2-May-1979
0000    71    ; 1-006 - Remove all entry points that need object time formatting,
0000    72    ;          putting them in FOR$ENTRY_OBJ so that we can arrange to
0000    73    ;          load the format compiler only when it is needed.
0000    74    ;          JBS 26-JUN-1979
0000    75    ; 1-007 - Remove entry point FOR$ENCODE_MF; we will code a new module
0000    76    ;          for it and FOR$$IO_BEG, to see how much I/O initiation time
0000    77    ;          improves.  JBS 02-JUL-1979
0000    78    ; 1-008 - Do likewuse for FOR$READ_DU and FOR$WRITE_DU.  JBS 03-JUL-1979
0000    79    ; 1-009 - Remove all entry points and add FOR$WRITE_SO;  each entry
0000    80    ;          point gets its own module do we can selectively load
0000    81    ;          the necessary UDF and REC modules.  JBS 09-JUL-1979
0000    82    ; 1-010 - New parameter format for FOR$$IO_BEG.  SBL 5-Dec-1979
0000    83    ; 1-011 - Change BRW FOR$$IO_BEG+2 to JMP G^FOR$$IO_BEG+2.  JAW 21-Feb-1981
```

```
         0000       85              .SBTTL  DECLARATIONS
         0000       86
         0000       87 ;
         0000       88 ; INCLUDE FILES:
         0000       89 ;
         0000       90
         0000       91         $FORPAR                            ; Define inter-module FORTRAN symbols
         0000       92         $ISBDEF                            ; Define statement type symbols
         0000       93
         0000       94 ;
         0000       95 ; EXTERNAL SYMBOLS:
         0000       96 ;
         0000       97
         0000       98         .DSABL  GBL                        ; Declare all external symbols
         0000       99         .EXTRN  FOR$$IO_BEG                ; common I/O statement processing
         0000      100 ;+
         0000      101 ; The following references are to make sure the necessary UDF and REC
         0000      102 ; modules are loaded.  These are the routines which are called through
         0000      103 ; the dispatch tables in FOR$$DISPAT.
         0000      104 ;-
         0000      105         .EXTRN  FOR$$UDF_WF0, FOR$$UDF_WF1, FOR$$UDF_WF9
         0000      106         .EXTRN  FOR$$REC_WSF0, FOR$$REC_WSF1, FOR$$REC_WSF9
         0000      107
         0000      108 ;+
         0000      109 ; The following reference makes sure the format compiler is loaded.
         0000      110 ;-
         0000      111         .EXTRN  FOR$$FMT_COMPIL
         0000      112
         0000      113 ;
         0000      114 ; MACROS:
         0000      115 ;
         0000      116 ;     NONE
         0000      117 ;
         0000      118 ; PSECT DECLARATIONS:
         0000      119 ;
         0000      120
      00000000      121         .PSECT  _FOR$CODE PIC,USR,CON,REL,LCL,SHR,EXE,RD,NOWRT,LONG
         0000      122
         0000      123 ;
         0000      124 ; EQUATED SYMBOLS:
         0000      125 ;
         0000      126
         0000      127
         0000      128 ;
         0000      129 ; OWN STORAGE:
         0000      130 ;
         0000      131 ;     NONE
         0000      132 ;
```

```
                        0000  134              .SBTTL  FOR$WRITE_SO - WRITE Sequential OBJECT-FORMATTED
                        0000  135
                        0000  136  ;++
                        0000  137  ; FUNCTIONAL DESCRIPTION:
                        0000  138  ;
                        0000  139  ;        Initialize the FORTRAN I/O system to perform
                        0000  140  ;        a write sequential OBJECT-FORMATTED I/O statement.
                        0000  141  ;
                        0000  142  ; CALLING SEQUENCE:
                        0000  143  ;
                        0000  144  ;        CALL FOR$WRITE_SO (unit.rl.v, format_adr.rt.r
                        0000  145  ;                 [, err_adr.j.r [, end_adr.j.F]])
                        0000  146  ;
                        0000  147  ; INPUT PARAMETERS:
                        0000  148  ;
                        0000  149  ;        unit.rl.v                  logical unit number
                        0000  150  ;        format_adr.rt.r            format string (needs compilation)
                        0000  151  ;        [err_adr.j.r]              optional ERR= address
                        0000  152  ;        [end_adr.j.r]              optional END= address
                        0000  153  ;
                        0000  154  ; IMPLICIT INPUTS:
                        0000  155  ;
                        0000  156  ;        NONE except those used by FOR$$IO_BEG.
                        0000  157  ;
                        0000  158  ; OUTPUT PARAMETERS:
                        0000  159  ;
                        0000  160  ;        NONE
                        0000  161  ;
                        0000  162  ; IMPLICIT OUTPUTS:
                        0000  163  ;
                        0000  164  ;        NONE except those left by FOR$$IO_BEG.
                        0000  165  ;
                        0000  166  ; COMPLETION CODES:
                        0000  167  ;
                        0000  168  ;        NONE
                        0000  169  ;
                        0000  170  ; SIDE EFFECTS:
                        0000  171  ;
                        0000  172  ;        NONE except those of FOR$$IO_BEG.
                        0000  173  ;
                        0000  174  ;--
                        0000  175
                  0000' 0000  176  FOR$WRITE_SO::  .MASK   FOR$$IO_BEG
    50   0101 8F  3C' 0002  177          MOVZWL  #ISB$K_ST_TY_WSF+                     -
                        0007  178                  <1@FOR$V_OBJ_FMT>, R0    ; Statement type
    00000002'GF  17    0007  179          JMP     G^FOR$$IO_BEG+2          ; branch past call mask
                        000D  180
                        000D  181
                        000D  182          .END
```

F 11

FOR$WRITE_SO   - entry point for FORTRAN WRITE SEQUENTI 16-SEP-1984 00:07:06 VAX/VMS Macro V04-00   Page 5
Symbol table   6-SEP-1984 11:02:14 [FORRTL.SRC]FORWRITSO.MAR;1   (4)

```
FOR$$FMT_COMPIL                  ********  X    00
FOR$$IO_BEG                      ********  X    00
FOR$$REC_WSF0                    ********  X    00
FOR$$REC_WSF1                    ********  X    00
FOR$$REC_WSF9                    ********  X    00
FOR$$UDF_WF0                     ********  X    00
FOR$$UDF_WF1                     ********  X    00
FOR$$UDF_WF9                     ********  X    00
FOR$V_OBJ_FMT         = 00000008
FOR$WRITE_SO            00000000 RG        01
ISB$K_ST_TY_WSF       = 00000001
```

```
                          +-------------------+
                          ! Psect synopsis !
                          +-------------------+
```

```
PSECT name                     Allocation          PSECT No.  Attributes
----------                     ----------          ---------  ----------
.  ABS  .                      00000000 (    0.)  00 (   0.)  NOPIC  USR   CON   ABS   LCL  NOSHR  NOEXE  NORD   NOWRT NOVEC BYTE
_FOR$CODE                      0000000D (   13.)  01 (   1.)   PIC   USR   CON   REL   LCL   SHR    EXE    RD    NOWRT NOVEC LONG
```

```
                    +-------------------------------+
                    ! Performance indicators !
                    +-------------------------------+
```

```
Phase                   Page faults    CPU Time        Elapsed Time
-----                   -----------    --------        ------------
Initialization                  31     00:00:00.10     00:00:00.91
Command processing             130     00:00:00.63     00:00:03.63
Pass 1                         126     00:00:01.22     00:00:04.44
Symbol table sort                0     00:00:00.19     00:00:00.45
Pass 2                          47     00:00:00.45     00:00:01.79
Symbol table output              3     00:00:00.02     00:00:00.02
Psect synopsis output            2     00:00:00.03     00:00:00.32
Cross-reference output           0     00:00:00.00     00:00:00.00
Assembler run totals           341     00:00:02.64     00:00:11.56
```

The working set limit was 1050 pages.
6711 bytes (14 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 188 non-local and 0 local symbols.
182 source lines were read in Pass 1, producing 8 object records in Pass 2.
9 pages of virtual memory were used to define 2 macros.

```
                    +-------------------------------+
                    ! Macro library statistics !
                    +-------------------------------+
```

```
Macro library name                      Macros defined
------------------                      --------------
-$255$DUA28:[FORRTL.OBJ]FORRTL.MLB;1          2
-$255$DUA28:[SYSLIB]STARLET.MLB;2             0
TOTALS (all libraries)                        2
```

183 GETS were required to define 2 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:FORWRITSO/OBJ=OBJ$:FORWRITSO MSRC$:FORWRITSO/UPDATE=(ENH$:FORWRITSO)+LI

HELP

FORWRITDF
LIS

VMSHELP
MAP

HLD
MDL

FORWRITDU
LIS

FORVECTOR
LIS

FORWRITIL
LIS

FORWRITSF
LIS

HLD

HLDMACROS
MAR

FORVM
LIS

FORWRITSN
LIS

HLD
MAP

FORWRITSU
LIS

RSXDEFS
MAR

FORUNLOCK
LIS

FORWRITDO
LIS

HELP
LIS

FORWRITIF
LIS

FORWRITIO
LIS

HLDCSTA
LIS

FORUDFWU
LIS

FORUNDERF
LIS

FORWRITSL
LIS

FORWRITSO
LIS